Q1. Write VHDL code for the following:

1) D-flip-flop

D Flip Flop Entity

D_FLIP_FLOP.vhd | D_flip_flop_test_bench.vhd | D_flip_flop_test_bench.vhd

```
1
2   library ieee;
3   use ieee.std_logic_1164.all;
4
5   entity D_FLIP_FLOP is
6       port(
7           q: out std_logic;
8           clk: in std_logic;
9           synchronous_reset: in std_logic;
10          d: in std_logic
11          );
12
13  end D_FLIP_FLOP;
14
15  architecture behavioral of D_FLIP_FLOP is
16
17  begin
18
19      clocked_process: process(clk)
20      begin
21          if (rising_edge(clk)) then
22              if (synchronous_reset = '1') then
23                  q <= '0';
24              else
25                  q <= d;
26              end if;
27          end if;
28      end process clocked_process;
29  end behavioral;
30
```

D Flip Flop Test Bench

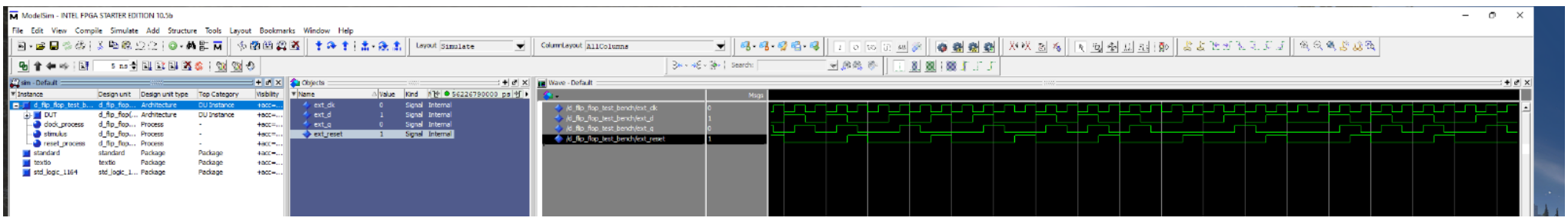
C:\Users\musil\OneDrive - Ohio University\First Year\Winter\Design of Digital Circuits\Module 9\D_flip_flop_modelSim

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

D_FLIP_FLOP.vhd | D_flip_flop_test_bench.vhd

```vhdl
-- Mark Musil
-- This file describes a D flip flop and runs a few tests on it.

library ieee;
use ieee.std_logic_1164.all;

entity D_flip_flop_test_bench is
end entity;

architecture behavior of D_flip_flop_test_bench is

signal ext_d : std_logic := '1';
signal ext_q : std_logic;
signal ext_reset, ext_clk: std_logic := '0';

component D_FLIP_FLOP port(
                        clk, d, synchronous_reset : in std_logic;
                        q : out std_logic
                        ); end component;

begin

--  ext_clk <= NOT(ext_clk) after 5 ns;
--  ext_d <= NOT(ext_d) after 30 ns;
--  ext_reset <= NOT(ext_reset) after 50 ns;

    DUT: D_FLIP_FLOP port map(
                        clk => ext_clk,
                        d => ext_d,
                        synchronous_reset => ext_reset,
                        q => ext_q);

    clock_process: process
    begin
        ext_clk <= '0';
        wait for 5 ns;
        ext_clk <= '1';
        wait for 5 ns;
    end process;

    stimulus: process
    begin
        wait for 10 ns;
        ext_d <= '1';
        wait for 10 ns;
        ext_d <= '0';
        wait for 10 ns;
        ext_d <= '1';
        wait for 10 ns;
        ext_d <= '0';
        wait for 10 ns;
        ext_d <= '0';
        wait for 10 ns;
        ext_d <= '1';
        wait for 10 ns;
        ext_d <= '0';
    end process;

    reset_process: process
    begin
        wait for 30 ns;
        ext_reset <= '1';
        wait for 10 ns;
        ext_reset <= '0';
    end process;
end architecture behavior;
```

Simulation



2) T-flip-flop

T Flip Flop Entity

C:\blog\Tutorials\modelSim\t_flip_flop\t_flip_flop.vhd - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

t_flip_flop.vhd | t_flip_flop_test_bench.vhd

```
1
2   library ieee;
3   use ieee.std_logic_1164.all;
4
5   entity t_flip_flop is port (
6           clk, t, asynch_reset : in std_logic;
7           q : out std_logic
8           );
9       end t_flip_flop;
10
11  architecture letsBehave of t_flip_flop is
12
13  begin
14
15      t_flip_flop_process: process(clk, asynch_reset) begin
16          if(asynch_reset = '1') then
17              q <= '0';
18          elsif rising_edge(clk) then
19              if(t = '1') then
20                  q <= NOT(q);
21              end if;
22          end if;
23      end process t_flip_flop_process;
24  end letsBehave;
```

T Flip Flop Testbench

t_flip_flop.vhd | t_flip_flop_test_bench.vhd

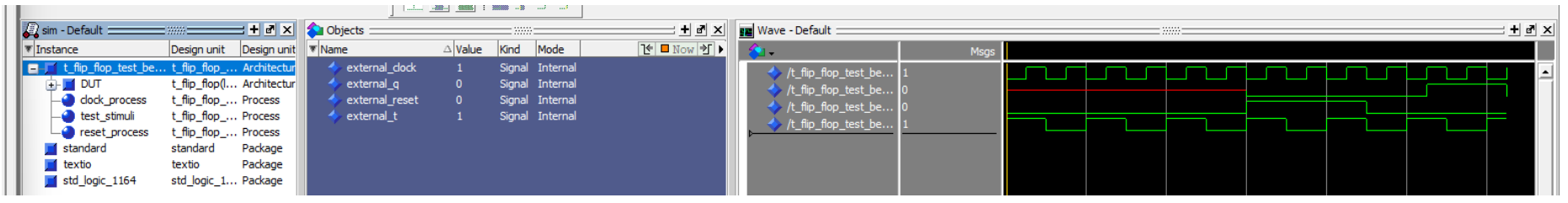
```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity t_flip_flop_test_bench is
5  end entity;
6
7  architecture pleaseBehave of t_flip_flop_test_bench is
8
9  signal external_t : std_logic := '1';
10 signal external_q : std_logic;
11 signal external_reset, external_clock: std_logic := '0';
12
13 component t_flip_flop port (
14         clk, t, asynch_reset : in std_logic;
15         q : out std_logic
16         ); end component;
17 begin
18
19     DUT: t_flip_flop port map(
20         clk => external_clock,
21         t => external_t,
22         asynch_reset => external_reset,
23         q => external_q);
24
25     clock_process: process
26     begin
27         external_clock <= '0';
28         wait for 5 ns;
29         external_clock <= '1';
30         wait for 5 ns;
31     end process;
32
33     test_stimuli: process
34     begin
35         wait for 10 ns;
36         external_t <= '0';
37         wait for 10 ns;
38         external_t <= '1';
39     end process;
40
41     reset_process: process
42     begin
43         wait for 60 ns;
44         external_reset <= '1';
45         wait for 30 ns;
46         external_reset <= '0';
47     end process;
48 end architecture pleaseBehave;
```

Mark Musil
February 18, 2022

Design of Digital Circuits
Module 9 Homework

Ohio University

T Flip Flop simulation



3) 8-bit wide serial shift register with D-flip-flops

D Flip-Flop

```
Ln#
 1
 2   library ieee;
 3   use ieee.std_logic_1164.all;
 4
 5   entity D_FLIP_FLOP is
 6           port(
 7                   q: out std_logic;
 8                   clk: in std_logic;
 9                   synchronous_reset: in std_logic;
10                   d: in std_logic
11                   );
12
13   end D_FLIP_FLOP;
14
15   architecture behavioral of D_FLIP_FLOP is
16
17   begin
18
19           clocked_process: process(clk)
20           begin
21                   if (rising_edge(clk)) then
22                           if (synchronous_reset = '1') then
23                                   q <= '0';
24                           else
25                                   q <= d;
26                           end if;
27                   end if;
28           end process clocked_process;
29   end behavioral;
30
```

Eight-bit shift register

TB_eight_bit_shift_register.vhd | t_flip_flop.vhd | t_flip_flop.vhd | D_eight_bit_SR.vhd

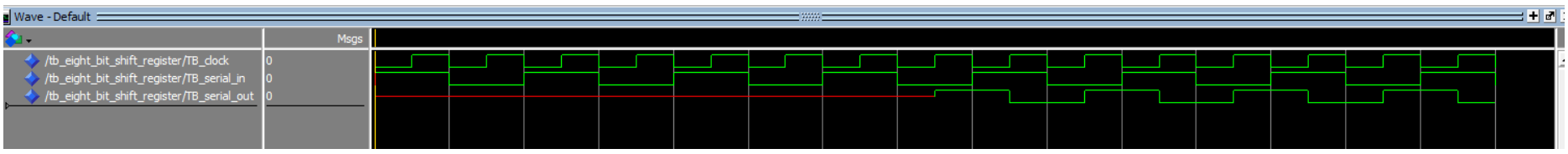
```
2   library ieee;
3   use ieee.std_logic_1164.all;
4
5   entity D_eight_bit_SR is port(
6       clock: in std_logic;
7       serial_in: in std_logic;
8       serial_out: out std_logic
9       );
10      end D_eight_bit_SR;
11
12  architecture behavioral of D_eight_bit_SR is
13
14  signal internal_bus : std_logic_vector(6 downto 0);
15
16  component D_FLIP_FLOP port(
17                          clk: in std_logic;
18                          d: in std_logic;
19                          q: out std_logic
20                          ); end component;
21
22  begin
23
24      D_FLIP_FLOP1: D_FLIP_FLOP
25          port map(
26                  clk => clock,
27                  d => serial_in,
28                  q => internal_bus(0)
29                  );
30
31      D_FLIP_FLOP2: D_FLIP_FLOP
32          port map(
33                  clk => clock,
34                  d => internal_bus(0),
35                  q => internal_bus(1)
36                  );
37
38      D_FLIP_FLOP3: D_FLIP_FLOP
39          port map (
40                  clk => clock,
41                  d => internal_bus(1),
42                  q => internal_bus(2)
43                  );
44
45
46      D_FLIP_FLOP4: D_FLIP_FLOP
47          port map(
48                  clk => clock,
49                  d => internal_bus(2),
50                  q => internal_bus(3)
51                  );
52
53      D_FLIP_FLOP5: D_FLIP_FLOP
54          port map(
55                  clk => clock,
56                  d => internal_bus(3),
57                  q => internal_bus(4)
58                  );
59
60      D_FLIP_FLOP6: D_FLIP_FLOP
61          port map (
62                  clk => clock,
63                  d => internal_bus(4),
64                  q => internal_bus(5)
65                  );
66      D_FLIP_FLOP7: D_FLIP_FLOP
67          port map (
68                  clk => clock,
69                  d => internal_bus(5),
70                  q => internal_bus(6)
71                  );
72
73      D_FLIP_FLOP8: D_FLIP_FLOP
74          port map (
75                  clk => clock,
76                  d => internal_bus(6),
77                  q => serial_out
78                  );
79
80  end behavioral;
81
```

Test bench

C:/Users/musil/OneDrive - Ohio University/First Year/Winter/Design of Digital Circuits/Module 9/8_b

```
 1  library ieee;
 2  use ieee.std_logic_1164.all;
 3
 4  entity TB_eight_bit_shift_register is
 5  end entity;
 6
 7
 8  architecture behavioral of TB_eight_bit_shift_register is
 9
10  signal TB_clock: std_logic := '0';
11  signal TB_serial_in: std_logic := '0';
12  signal TB_serial_out: std_logic;
13
14
15  component D_eight_bit_SR port (
16                  clk: in std_logic;
17                  serial_in: in std_logic;
18                  serial_out: out std_logic
19                  );
20  end component;
21
22  begin
23
24          DUT: D_eight_bit_SR port map(
25                  clk => TB_clock,
26                  serial_in => TB_serial_in,
27                  serial_out => TB_serial_out
28                  );
29
30          
31          clock_process: process
32
33          begin
34                  TB_clock <= '0';
35                  wait for 5 ns;
36                  TB_clock <='1';
37                  wait for 5 ns;
38          end process;
39
40          test_stimuli: process
41          begin
42                  TB_serial_in <= '1';
43                  wait for 10 ns;
44                  TB_serial_in <= '0';
45                  wait for 10 ns;
46                  TB_serial_in <= '1';
47                  wait for 10 ns;
48                  TB_serial_in <= '0';
49                  wait for 10 ns;
50                  TB_serial_in <= '1';
51                  wait for 10 ns;
52                  TB_serial_in <= '0';
53                  wait for 10 ns;
54                  TB_serial_in <= '1';
55                  wait for 10 ns;
56                  TB_serial_in <= '0';
57                  wait for 10 ns;
58          end process;
59
60  end behavioral;
```

Simulation Results



Q2. Do question 4.13 (chapter 4).

4.13. Find the errors in the following code:

```
architecture has_errors of design
begin
p1: process
    begin
        if clk'event and clk = '1' then
            q <= a or b and c;
        end if;
    end;
```

This block of VHDL is missing the following

- Entity declaration
    - Associated port declarations
- Process should be ended explicitly
- Has_errors should be ended explicitly
- The code should be declared with the clock in its sensitivity list

Correct Version

*C:\Users\musil\OneDrive - Ohio University\First Year\Winter\Design of Digital

File Edit Search View Encoding Language Settings Tools Macro R

TB_eight_bit_shift_register.vhd | D_eight_bit_SR.vhd | 4p13.vhd

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity my_entity is
port (
    a: in std_logic;
    b: in std_logic;
    c: in std_logic;
    clk: in std_logic;
    q: out std_logic
    ); end my_entity;

architecture has_errors of my_entity is

 begin

    p1: process(clk)

        begin
            if clk='1' then
                q <= ((a or b) and c);
            end if;
    end process p1;

end has_errors;
```

Q3. Do question 4.15 (chapter 4).

4.15. Do all processes require sensitivity lists? Can you declare a clocked process without a sensitivity list?

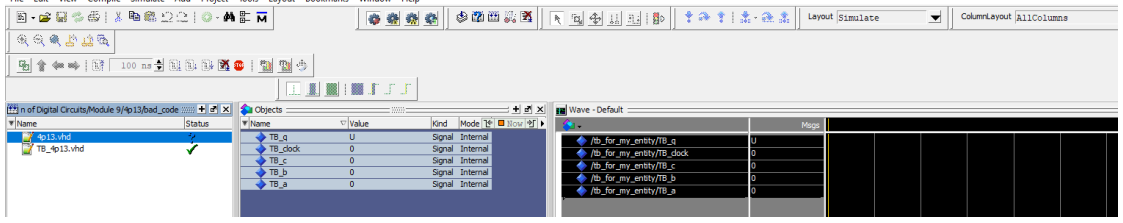
No. Without a sensitivity list the compiler cannot determine when to run the process. I attempted to run the fixed code from the previous problem without using a sensitivity list and the simulator could not start the simulation. See the simulator window below.

Without sensitivity list

TB_eight_bit_shift_register.vhd | D_eight_bit_SR.vhd | 4p13.vhd

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity my_entity is
5  port (
6       a: in std_logic;
7       b: in std_logic;
8       c: in std_logic;
9       clk: in std_logic;
10      q: out std_logic
11      ); end my_entity;
12
13 architecture has_errors of my_entity is
14
15
16
17 begin
18
19      p1: process
20
21          begin
22              if (clk'event and clk='1') then
23                  q <= ((a or b) and c);
24              end if;
25      end process p1;
26
27 end has_errors;
```

Simulation without sensitivity list



However, when I added a sensitivity list, as shown in the code snippet below, the simulator ran well (also shown)

With sensitivity list

TB_eight_bit_shift_register.vhd | D_eight_bit_SR.vhd | 4p13.vhd

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity my_entity is
5   port (
6       a: in std_logic;
7       b: in std_logic;
8       c: in std_logic;
9       clk: in std_logic;
10      q: out std_logic
11      ); end my_entity;
12
13  architecture has_errors of my_entity is
14
15  begin
16
17      p1: process(clk)
18
19          begin
20              if clk='1' then
21                  q <= ((a or b) and c);
22              end if;
23      end process p1;
24
25  end has_errors;
26
27
```

Simulation with sensitivity list