Testbench Automation

Last updated 5/14/20

Testbench Automation

These slides describe how to create automated testbench verification code

Upon completion: You should be able to write testbench code that does not require visual verification of waveforms

Testbench Automation

- Manual Testbench
  - Results checked by hand
    - Simulation results in waveforms or output file
    - Check expected results to actual results by hand

Testbench Automation

- Automated Testbench
  - Results checked automatically
    - Code results into the test bench
      - Enumerated
      - Calculated
    - Check expected results to actual results in the testbench

Testbench Automation

- Automated Testbench
  - Design Verification – enumerated results

```
--
--  counter_nbit_ud_tb1.vhdl
--
--  by: johnsontimoj
--
--  created: 6/28/2016
--
--  version: 0.0
--
-----------------------------------
-----------------------------------
--
--  test bench for 3 bit counter
--
--  using enumerated results
--
--  inputs: none
--
--  outputs: count
--
--  sim run time = 370 ns
--
-----------------------------------

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

entity counter_nbit_ud_tb1 is
end entity;
```

```
architecture testbench of counter_nbit_ud_tb1 is
   constant per:     time  := 20 ns;
   constant t_delay: time  := 5 ns;
   constant t_reset: time  := per*2;

   -----------------------------------
   --  3 bit counter prototype
   -----------------------------------
   component counter_nbit_ud is
   port  (  i_clk:      in std_logic;
            i_rstb:     in std_logic;
            i_dir:      in std_logic;     -- dir=0 is up
            o_count:    out std_logic_vector(3-1 downto 0)
         );
   end component;
   -----------------------------------

   signal clk_tb:    std_logic := '0';
   signal rstb_tb:   std_logic;
   signal dir_tb:    std_logic := '0';    -- dir=0 is up
   signal count_tb:  std_logic_vector(3-1 downto 0) := (others => '0');

   signal count_chk: std_logic_vector(3-1 downto 0);
```

```
begin
   -----------------------------------
   --  Device Under Test
   -----------------------------------
   dut: counter_nbit_ud port map
      (  i_clk    => clk_tb,
         i_rstb   => rstb_tb,
         i_dir    => dir_tb,
         o_count  => count_tb
      );
   -----------------------------------

   -- Clock Process
   clk: process
      begin
         wait for per/2;
         clk_tb <= not clk_tb;
   end process;

   -- Reset process (active low)
   rstb: process
      begin
         rstb_tb <= '0';
         wait for t_reset;
         rstb_tb <= '1';
         wait;       -- only executes once
   end process;

   -- Direction process
   dir: process
      begin
         wait for t_reset;
         wait for per*8;
         dir_tb <= not dir_tb;
      end process;
```

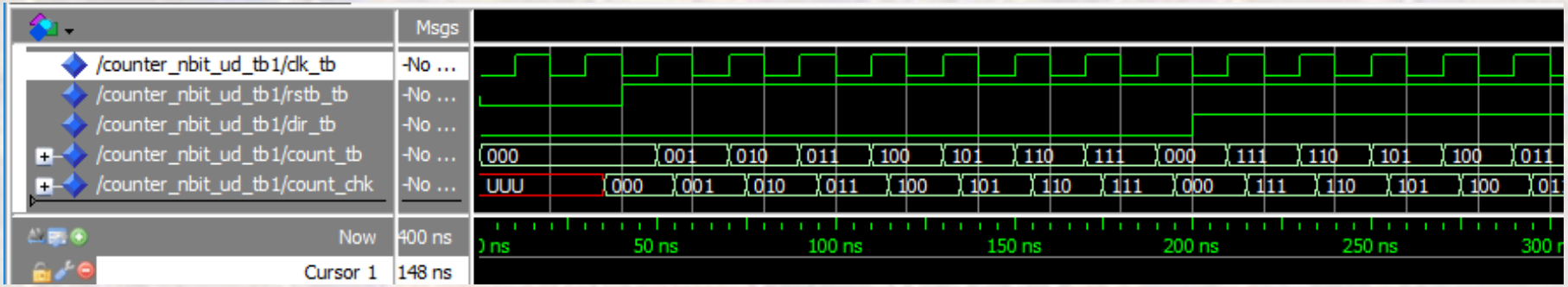
```
   -----------------------------------
   --  generate expected results
   -----------------------------------
   count_chk <=   "000" after t_reset-per/2 + per*0 + t_delay,
                  "001" after t_reset-per/2 + per*1 + t_delay,
                  "010" after t_reset-per/2 + per*2 + t_delay,
                  "011" after t_reset-per/2 + per*3 + t_delay,
                  "100" after t_reset-per/2 + per*4 + t_delay,
                  "101" after t_reset-per/2 + per*5 + t_delay,
                  "110" after t_reset-per/2 + per*6 + t_delay,
                  "111" after t_reset-per/2 + per*7 + t_delay,
                  "000" after t_reset-per/2 + per*8 + t_delay,
                  "111" after t_reset-per/2 + per*9 + t_delay,
                  "110" after t_reset-per/2 + per*10 + t_delay,
                  "101" after t_reset-per/2 + per*11 + t_delay,
                  "100" after t_reset-per/2 + per*12 + t_delay,
                  "011" after t_reset-per/2 + per*13 + t_delay,
                  "010" after t_reset-per/2 + per*14 + t_delay,
                  "001" after t_reset-per/2 + per*15 + t_delay,
                  "000" after t_reset-per/2 + per*16 + t_delay,
                  "111" after t_reset-per/2 + per*17 + t_delay;
   -----------------------------------
```

```
   -----------------------------
   --  check actual results
   -----------------------------
   process
   begin
      -- delay for reset
      if (now < per*2) then
         wait for per/2;
      -- run to end of sim
      elsif (now < 20*per) then
         assert (count_tb = count_chk)
            report   "ERROR at t= " & time'image(now) &
                     ", count_tb = " & integer'image(to_integer(unsigned(count_tb))) &
                     ", count_chk = " & integer'image(to_integer(unsigned(count_chk)))
            severity failure;
      else
         assert (false)
            report "no errors found at t=" & time'image(now)
            severity note;
      end if;
      wait for per;
   end process;

end architecture;
```

Testbench Automation

- Automated Testbench
  - Design Verification – enumerated results

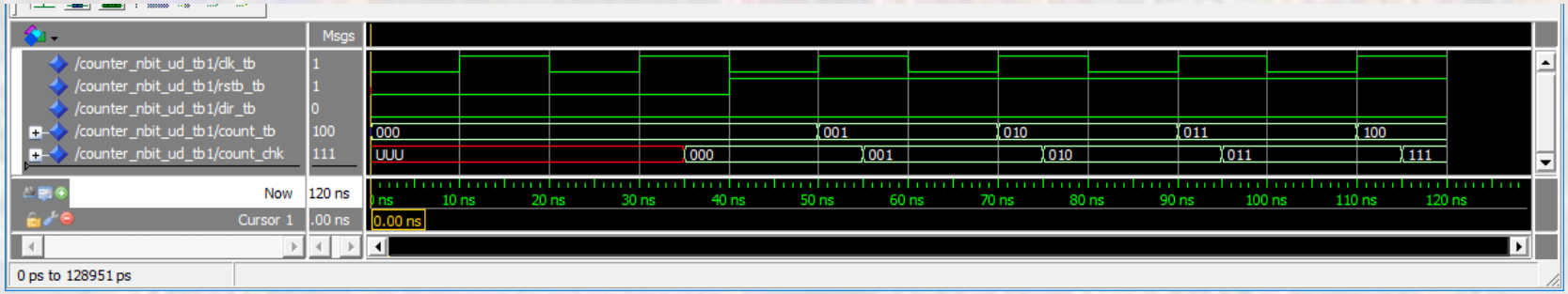



```
#
# add wave *
# view structure
# .main_pane.structure.interior.cs.body.struct
# view signals
# .main_pane.objects.interior.cs.body.tree
# run 400 ns
# ** Note: no errors found at t=400000 ps
#    Time: 400 ns  Iteration: 0  Instance: /counter_nbit_ud_tb1

VSIM 2>
```

Testbench Automation

- Automated Testbench
  - Design Verification – enumerated results
    - Changed expected result 4 from 3 to 7




```
#
# add wave *
# view structure
# .main_pane.structure.interior.cs.body.struct
# view signals
# .main_pane.objects.interior.cs.body.tree
# run 400 ns
# ** Failure: ERROR at t= 120000 ps, count_tb = 4, count_chk = 7
#    Time: 120 ns  Iteration: 0  Process: /counter_nbit_ud_tb1/line__120 File: D:/GDrive/MSOE/19_Q1_EE3921/Projects/Class_Examples/counter_nbit_ud_tb1.vhdl
# Break in Process line__120 at D:/GDrive/MSOE/19_Q1_EE3921/Projects/Class_Examples/counter_nbit_ud_tb1.vhdl line 129

VSIM 2>
```

Testbench Automation

- Automated Testbench
  - Design Verification – calculated results

```
begin
   ----------------------------------
   --  Device Under Test
   ----------------------------------
   dut: counter_nbit_ud port map
      (  i_clk    => clk_tb,
         i_rstb   => rstb_tb,
         i_dir    => dir_tb,
         o_count  => count_tb
      );
   ----------------------------------

   -- Clock Process
   clk: process
      begin
         wait for per/2;
         clk_tb <= not clk_tb;
   end process;

   -- Reset process (active low)
   rstb: process
      begin
         rstb_tb <= '0';
         wait for t_reset;
         rstb_tb <= '1';
         wait;          -- only executes once
   end process;

   -- Direction process
   dir: process
      begin
         wait for t_reset;
         wait for per*8;
         dir_tb <= not dir_tb;
      end process;
```

```
----------------------------------
--
--  counter_nbit_ud_tb2.vhdl
--
--  by: johnsontimoj
--
--  created: 6/28/2016
--
--  version: 0.0
--
----------------------------------
----------------------------------
--
--  test bench for 3 bit counter
--
--  using calculated results
--
--  inputs: none
--
--  outputs: count
--
--  sim run time = 370 ns
--
----------------------------------

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

entity counter_nbit_ud_tb2 is

end entity;
```

```
architecture testbench of counter_nbit_ud_tb2 is
   constant per:     time  := 20 ns;
   constant t_delay: time  := 5 ns;
   constant t_reset: time  := per*2;

   ----------------------------------
   --  3 bit counter prototype
   ----------------------------------
   component counter_nbit_ud is
   port  (  i_clk:      in std_logic;
            i_rstb:     in std_logic;
            i_dir:      in std_logic;      -- dir=0 is up
            o_count:    out std_logic_vector(3-1 downto 0)
         );
   end component;
   ----------------------------------

   signal clk_tb:    std_logic := '0';
   signal rstb_tb:   std_logic;
   signal dir_tb:    std_logic := '0';    -- dir=0 is up
   signal count_tb:  std_logic_vector(3-1 downto 0) := (others => '0');

   signal count_chk: std_logic_vector(3-1 downto 0);
```

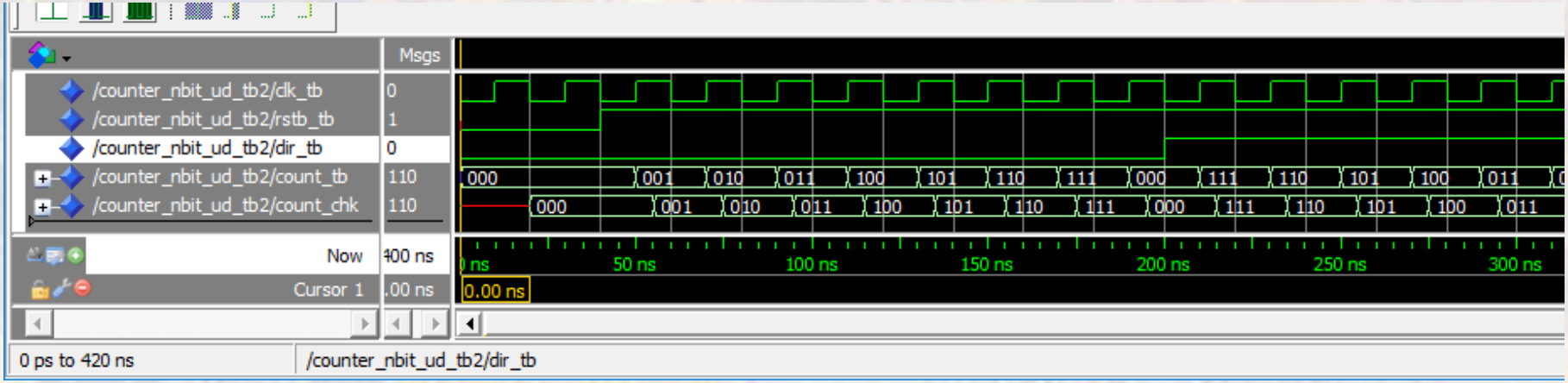
```
   ----------------------------------
   --  generate expected results
   ----------------------------------
   process
   begin
      if (rstb_tb = '0') then
         count_chk <= (others => '0');
         wait for per*3/4;
      else if (dir_tb = '0') then
            count_chk <= std_logic_vector(unsigned(count_chk) + 1);
         else
            count_chk <= std_logic_vector(unsigned(count_chk) - 1);
         end if;
      end if;
      wait for per;
   end process;
   ----------------------------------

   ----------------------------------
   --  check actual results
   ----------------------------------
   process
   begin
      -- delay for reset
      if (now < per*2.5) then
         wait for per/2;
      -- run to end of sim
      elsif (now < 20*per) then
         assert (count_tb = count_chk)
            report   "ERROR at t= " & time'image(now) &
                     ", count_tb = " & integer'image(to_integer(unsigned(count_tb))) &
                     ", count_chk = " & integer'image(to_integer(unsigned(count_chk)))
            severity failure;
      else
         assert (false)
            report "no errors found at t=" & time'image(now)
            severity note;
      end if;
      wait for per;
   end process;

end architecture;
```

Testbench Automation

- Automated Testbench
  - Design Verification – calculated results




```
# add wave *
# view structure
# .main_pane.structure.interior.cs.body.struct
# view signals
# .main_pane.objects.interior.cs.body.tree
# run 400 ns
# ** Note: no errors found at t=400000 ps
#    Time: 400 ns  Iteration: 0  Instance: /counter_nbit_ud_tb2

VSIM 2>
```

Now: 400 ns Delta: 2 | sim:/counter_nbit_ud_tb2